Table of Contents

1 PURPOSE AND BACKGROUND OF SYSTEM ......................................................... 3
  1.1 DEVELOPMENT PROCESS ........................................................................... 5
2 BACKGROUND ..................................................................................................... 5
3 DESCRIPTION OF SYSTEM ............................................................................... 6
4 ENVIRONMENT ASSUMPTIONS ....................................................................... 8
5 FUNCTIONAL REQUIREMENTS .......................................................................... 8
  5.1 GENERAL ...................................................................................................... 9
  5.2 COLLECTIONS ............................................................................................ 9
    5.2.1 General ................................................................................................ 9
    5.2.2 Collection creation .............................................................................. 10
    5.2.3 Collection maintenance ...................................................................... 10
    5.2.4 Collection deletion ............................................................................ 10
    5.2.5 Collection archiving .......................................................................... 10
  5.3 CONTENT ...................................................................................................... 11
    5.3.1 Root page ............................................................................................ 12
    5.3.2 Accessions register ............................................................................. 12
    5.3.3 Content creation (basic) ..................................................................... 12
    5.3.4 Content creation (supported) ............................................................... 13
    5.3.5 Content alteration .............................................................................. 13
    5.3.6 Content access .................................................................................. 14
    5.3.7 Content deletion ................................................................................ 15
  5.4 COMMUNITIES ............................................................................................. 15
    5.4.1 Community management .................................................................... 16
    5.4.2 Community creation .......................................................................... 16
    5.4.3 Community maintenance ................................................................... 17
    5.4.4 Community deletion .......................................................................... 17
    5.4.5 Joining communities .......................................................................... 17
    5.4.6 Leaving communities ........................................................................ 18
    5.4.7 Community membership maintenance .............................................. 19
  5.5 SYSTEM REQUIREMENTS .......................................................................... 20
    5.5.1 Security of contents .......................................................................... 20
    5.5.2 Preservation ....................................................................................... 20
6 NON-FUNCTIONAL REQUIREMENTS ................................................................. 21
  6.1 RESPONSE TIME ...................................................................................... 21
  6.2 NUMBER OF COLLECTIONS ....................................................................... 21
  6.3 NUMBER OF ACCESS COMMUNITIES .......................................................... 21
  6.4 NUMBER OF USERS .................................................................................. 21
  6.5 COLLECTION SIZE .................................................................................... 22
7 CHALLENGES AND DEVELOPMENT ISSUES ............................................... 22
## Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2007-07-09</td>
<td>Initial strawman</td>
</tr>
<tr>
<td>0.2</td>
<td>2007-12-07</td>
<td>Revised</td>
</tr>
<tr>
<td>1.0</td>
<td>2007-03-06</td>
<td>Revised</td>
</tr>
<tr>
<td>1.1</td>
<td>2007-03-20</td>
<td>Final version</td>
</tr>
</tbody>
</table>
1 Purpose and background of system

The Koorie Annotation System (KAS) will allow Koorie communities to annotate public records. It will also support Koorie communities to capture their own records, although this is not its primary purpose.

The ARC Linkage Trust and Technology project: building archival systems for Indigenous oral memory\(^1\) (T&T project) was developed in partnership between Monash University, Public Record Office Victoria (PROV), Koorie Heritage Trust Inc. (KHT), Victorian Koorie Records Taskforce and the Australian Society of Archivists, Indigenous Issues Special Interest Group. The premise behind the project is that whilst Koorie people value oral memory ‘there has been no extensive effort by Australian archival services to discover how to capture it and make it accessible to the relevant communities’ through time.’ [15] The project commenced in March 2004 and further details can be found at http://www.sims.monash.edu.au/research/eirg/trust/index.html.

The T&T project consisted of three stages:

- **Stage 1 – User Needs Analysis**
- **Stage 2 – Modelling Indigenous community-orientated archival services**
- **Stage 3 – Using information systems to build trusted archival systems**

Stages one and two of the T&T Project have revealed a need within the Koorie community to not merely archive oral memory, but to also bring together and annotate existing archival records. The draft report of the Trust and Technology Project identified the need for a holistic, community-based approach to Koorie archives (Outcome 4). A preliminary list of the features of such an approach would include:

- The ability to ‘bring records together virtually’ making dispersed records of a variety of forms and sources available via one gateway.
- A socio-technical model enabling relative, transaction based application of rights and obligations relating to custody, control, and access.
- A sound rights management environment to maximise community control over archives.
- Support for recovering Koorie knowledge from government records.
- Options for the storage and preservation of text-based records, photos, and audio records.
- The addressing of management and sustainability issues, including the need for suitably trained and experienced Indigenous staff, and developing and sustaining the necessary technical infrastructure and policy environment.

The vision for KAS is to provide a system where Koorie people can comment on, organise, store, and preserve digital records of their culture to serve their own needs. The KAS will allow communities to organise and comment on records (both entered into the system and held by other systems). All information in the KAS is presented in the form of web pages with the ability to include non-written content (images, sound, video). Consequently, like existing systems such as the Ara Irititja project [1, 6, 5], the KAS will have the capability for Koorie communities to repatriate information about their history back to the community, and to allow the community to create and store their own records.

The contents of a KAS could be original material (such as photographs, sound or video recordings, or text) or annotations about original material. Annotations describe or interpret external records or the contents of the KAS. For example, they could be descriptions, transcripts, indexes, or an arrangement of the material. Annotations refer to original material in the KAS, to original material held by other institutions, or to other annotations. Annotations need not be textual. They could be a sound or video clip.

---

\(^1\) Oral memory – for the T&T project oral memory refers to Australian Indigenous knowledge that originates and/or is reinterpreted as orally transmitted narratives or stories. For an extended discussion of the concept of oral memory as defined in this project and the related broad concept of Australian Indigenous oral history, encompassing the transmission of knowledge orally and the recording of Indigenous knowledge, see [4]
commenting on original material. A key feature is that the collection is expected to grow organically over time; for example a sound recording might not be transcribed for 50 years after being added to the collection.

The KAS is not intended to publish Koorie records to the general public over the web, nor is it intended to be run by a cultural institution to document ‘its’ records (although in both cases, the KAS could be used for both of these purposes).

A key focus of the KAS is management of the collection of information by the Koorie community themselves (i.e. the access protocols). It is clear from the data collected by the T&T project that if Koorie knowledge is to be recorded in any form, then Koorie communities will require that the KAS’s access control is flexible enough to allow multiple access models and requirements. The flexibility of access protocols applies to all aspects of KAS – data entry, editing and access to individual annotations. There has been no single access model identified from the data; it is possible that each annotation will consist of various access protocols – with the possibility of different mixes of individuals, groups, families, communities, and public having access to, or permission to edit, a single annotation. This type of access cannot be a simple binary function (grant/not grant access). It needs to be a system where different groups and individuals have access to different views of the annotation in accordance with the access protocols attached to the annotation. This may involve providing redacted views of content.

The community sets the policy for a collection, in particular who can create content, and who may view it. The community may construct the collection for individuals, for their own community, or for the broader community. This focus on control over who may add, modify, and access information distinguishes the KAS from typical ‘social networking’ Web systems such as the Wikipedia. Unlike other indigenous annotation systems (e.g. Koopman), access categories will be defined and controlled by the creators of information, not predefined by the system designers.

Because of the sensitivity attached to the need for strict access protocols, a KAS will need to be housed, maintained, and cared for by an organisation trusted by the Koorie community. The data from the T&T project indicates the community would prefer systems built to this specification to be run by Koorie community organisations than a government agencies. However, there is nothing in the specification that limits the type of organisation that could run a KAS. A KAS could even be run by an individual.

We consider trust to be a key issue in the successful deployment of the KAS. We believe that a key question will be: will users trust the system enough to add content? The following trust building mechanisms will be adopted:

- Community control of policies and content
- Multiple communities, not one community
- Severing the linkage with archival organisations (and government)

The KAS could be a first step for Victorian archives towards accommodating varying community needs and interaction with the collection. This could increase understanding, and trust between communities and archives. It is believed that online annotation systems, such as KAS, would be a method of interacting with archives that would be welcomed by all community groups, whilst enhancing the value of archival records. The benefits to archives and the Victorian community include many of those listed above for the Koorie community, but also include:

- An exciting extension to existing archives that does not interfere with their current regulations and systems;
- Provide a standard framework within which an archive could integrate a number of existing archival activities, such as the output of the volunteers program, and the finding aids produced by access services, and indexes and guides to its records;

---

2 This includes annotation records.
1.1 Development process

The KAS specification was primarily developed by a team at Public Record Office Victoria consisting of:

- Andrew Waugh
- Simon Flagg
- Daniel Wilksch
- Graeme Hairsine

In addition we wish to acknowledge the assistance of Rebecca French, Monash University, in carrying out a literature survey.

Drafts of the KAS were presented to a reference group that was established to validate and refine the specification. We wish to acknowledge the members of this reference group for their participation and efforts. The reference group consisted of:

- Jason Eades, Koorie Heritage Trust
- Simon Flagg, Public Record Office Victoria
- Antoinette Smith, Museum Victoria
- Lenny Tregonning, Koorie Heritage Trust
- Larry Walsh, Elder
- Diane Singh, Monash University
- Dr Graeme Johanson, Monash University
- Rebecca French, Monash University

We would particularly like to acknowledge Diane Singh’s assistance in organising the reference group meetings.

2 Background

Within indigenous communities there have been a number of projects to repatriate information back to the community and to allow simple annotation of the information. Within Australia, the most significant is the Ara Irititja project [1, 6, 5] of the Pitjantjatjara Council which was originally developed to locate, copy, record, and make accessible to the Anangu tjuta their cultural heritage for educational and community use. The archive is built around a database, and users can add, expand, or correct metadata associated with the objects. This system is popular in indigenous communities, however it is primarily a database. Resources must be described before they can be retrieved, restricting the addition of information, and there is no ability to organise the information to construct higher level descriptions of the collection, or link to external resources. The use of a database is common. For example Project Jukebox, coordinated by the University of Alaska, Fairbanks (http://uaf-db.uaf.edu/Jukebox/PJWeb/proguse.htm), is a program which has allowed Inuit communities in Alaska to form their own archives containing a range of information in a variety of formats.

The use of annotations and other collaborative features to enhance the online experiences to users of websites is beginning to be explored by cultural heritage institutions around the world. However, the work is in its early stages. Production or pilot systems exist at the The National Archive which provides a wiki to allow collaborative development of descriptions of its collection, Michigan Bentley Historical Library [17] to annotate the “Polar Bear Expedition Digital Collections” and the Bishop Museum in Hawaii [3] to annotate maps. The Steve Project [14] is a collaborative project developing social tagging for art museums. It allows users to register on the system, and then provide their own descriptions of particular art works. Initial results show that the descriptions that users create differ markedly from those created by curators. The Powerhouse Museum [11] and Smithsonian Photography [12] are developing their own similar in-house systems. University of Pennsylvania and Stanford University allow users to share and comment on particular library resources that they have bookmarked [2]. Similar proprietary sites that allow for these social tags or “folksonomies” are “del.icio.us” (http://del.icio.us/) and Flickr (http://www.flickr.com/). These forms of annotation allow collaborative development of resource descriptions and enable users to share...
bookmarks, rather than allowing high level annotations of digital objects. Most of these systems have a very limited ability to control access to the annotations, which can make them inappropriate in a indigenous context which typically emphasise strong control over who may know particular information.

One annotation system designed around the needs of indigenous communities, and which provides a strong access control mechanism is described in Hunter, Koopman and Sledge [9, 8]. They developed a system that allowed the description, annotation, and rights management of collections of mixed media digital and physical objects belonging to indigenous communities. Although the goal is similar to the system proposed in this paper, the approach is considerably different. The focus is on using metadata to describe, find, and control access to the objects. This has several limitations. First, it requires the development of a predefined metadata schema which fixes (and hence limits) the ways in which the object can be described and controlled. Second, it requires the creator to fill in a complex metadata form, both to describe the object and to set up the access control. Finally, it is only possible to search for content. The system specifically contained audio annotations to support cultural norms, and to aid with problems of computer literacy. It was noted that different annotations could present different viewpoints. A second subtle difference is that their system is described in the context where it is run by a cultural institution and the indigenous community can add annotations.

Many of these annotation systems are based on commercial or public domain software. Wikis are a widely used tool to allow communities of interest and practice to collaborate in the development of shared narratives. Wikis allow users to easily create new web pages about particular topics, while including features which permit users (often registered) to edit sections of the narrative [10]. Current wiki software has only a limited ability to allow for divergent narratives, or store audio or video content. They also do not address the long term management of the information held by the system. There are number of web annotation systems such as Annotea [16], Vannotea [13], and Annoty [7]. Annotea is client (browser) based and hence cannot be used to build collaborative annotations. Vannotea extends Annotea to provide a collaborative annotation space supporting the annotation of images, audio, and video by users with differing levels of access control. Annoty is similar.

### 3 Description of system

The primary purpose of KAS is to allow Koorie communities to provide alternative views to records held by existing archives through online annotations. Alternative views would include (but not be limited to):

- Corrections to existing records provided through Koorie oral memory;
- Family or community narratives and records that expand upon, or qualify, the content and context of existing archival records;
- Interpretations of individual documents, records, or series;
- Assistance in understanding or using the records, including the generation of finding aids;
- Linkages to other resources or records, which may be held by other organisations or individuals.

The system is designed around the needs of the Koorie communities of Victoria, as identified through stages one and two of the T&T project.

As outlined above, the key challenge of KAS will be the front and back access control protocols to the annotations; controlling the individual or group membership that creates and maintains particular annotations (the back or behind the scenes access), and the views that individuals, groups and communities will have access to (the front or public access). Preliminary analysis of the data from stage two of the T&T project indicates that a 'one-size-fits-all' approach to access control will not achieve the needs of different
Koorie community groups. The data indicates that communities and individuals who create the annotations will define the back and front access protocols for their record/s. This may mean that different community groups could have different views of the same underlying data (e.g. redacted views).

Whilst KAS is about placing access control in the hands of the annotators, there are concerns regarding quality control of entries. While cautious about the idea of monitoring, it is clear that no-one would want records to become the locale for inappropriate annotations and/or engagement. However there is an expectation that a minimum standard would need to be ensured.

A model of the proposed system is shown below.

The heart of the KAS is a collection. A collection contains content and has policies and communities. There are two types of community associated with each collection. The creator community adds and maintains the content contained in the collection. The creator community also sets and manages the policies for the community, in particular the policies that control the membership of the communities. The access communities can access the content. There may be several access communities associated with a collection, and each community can have different levels of access to the collection.

It is expected that there will be many separate autonomous collections held within a single KAS server. Each collection will be completely independent from other collections and will have its own communities, policies, and content.

Collections contain content which is the information that the creator community wishes to make available. All content is made available as web pages that can contain both text and non-textual (images, sound, and video). Conceptually, content is either original material or annotations. Original material could be oral memories (including video),

---

3 Community groups – family, community, tribe, nation and/or state.
4 Initial designs for the KAS had a separate administrator community that was responsible for setting and managing policies for the community. Feedback from the reference group was that they did not want a special group privileged with setting and managing policies. Consequently, the administrator and creator groups were combined.
photographs, or written material. Annotations assist users in using the original material. Annotations could be

- Commentary. This type of annotation includes additional information, corrections, clarifications, explanations, interpretations, or transcripts. The commentary could be about original material held in the collection, original material held by other archival systems, or even other pieces of commentary.
- Arrangements. This type of annotation is an arrangement of links to other content.
- Finding aids. This type of annotation includes descriptions, summaries, transcripts, and indexes.

In practice, the distinction between original material and annotations is not clear cut. An annotation is a piece of original material. It is a record created by a creator, and may contain supporting information (e.g. images or recordings).

The content referred to by the annotations may be within the collection, or outside the collection (e.g. in a formal government archive). In an ideal world, the content would be available on the web, but an annotation could refer to content that is only available on physically.

Content can be a mixture of different modes:

- Textual. By textual we mean typed information held by a computer, for example, web pages. In practice, textual pages will be implemented as web pages and may contain links to other content (both within and outside the system).
- Written. We make a distinction between textual material (i.e. where the characters are entered into a computer), and written material. The later is written material that can be read by a person and could be images of handwritten or typed documents.
- Photographic.
- Aural (i.e. audio).
- Video.

4 Environment assumptions

The KAS allows creators to comment on records held by archives (and other organisations). In order to do so we assume that:

- Significant amounts of records will be made available on the web. This assumption may not be warranted. Due to privacy, and the sensitive nature of Koorie records, these records might not be digitised, or if digitised, may not be available on the web.
- Web records are made available with a persistent URI. This means that references to external web pages will not break due to URLs being changed. If this assumption is not warranted, creators may not create annotation pages as the links could be too liable to break.

5 Functional requirements

This section details the functional requirements of the KAS. These are things which the KAS must do.

The requirements are divided into three categories. Requirements that the KAS must satisfy have a white background. Requirements with a light yellow background are ‘should’ requirements; those which the KAS should satisfy, but which might not be implemented if they are too expensive to implement. Requirements with a light orange background are ‘may’ requirements; those which we would like the KAS to satisfy, but which are expected to be complex or expensive to implement.
5.1 General

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The KAS must support the role of an administrator. The administrator operates the KAS and has the responsibilities as defined in the following requirements.</td>
</tr>
<tr>
<td>2.</td>
<td>The KAS must support an audit log. Events to be logged are specified below. Each event is to include the date/time, the identity of the user requesting the event, and the details of the event. The audit log is to be available to members of the creator community on a page linked from the root page. The default sort order is to be latest event to earliest event. Users will be able to reverse the sort order, and sort by user.</td>
</tr>
</tbody>
</table>

5.2 Collections

This section details the requirements concerned with creating, deleting, and managing collections within the KAS.

The heart of the KAS is a collection. A collection contains content and has associated policies and communities. Content is the information contained in the collection. Policies control how the collection operates, for example, who may access the content, and how content will appear to people accessing it. Communities are groups of people who may create or access the content.

A collection can be thought of as a web site.

It is expected that a single KAS server will hold many autonomous collections. Each collection on a server will be completely independent from other collections and will have its communities, policies, and content. To accessors, each collection will appear to be a separate web site. In particular, it will not be possible to search the content of all the collections except by using an external search engine such as Google.

5.2.1 General

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>The KAS must support multiple collections. Collections must be independent (that is a change in one collection to any aspect must not affect any other collection).</td>
</tr>
<tr>
<td>4.</td>
<td>Each collection will have a creator community</td>
</tr>
<tr>
<td>5.</td>
<td>Each collection will have one or more access communities</td>
</tr>
<tr>
<td>6.</td>
<td>Each collection will have a root page</td>
</tr>
<tr>
<td>7.</td>
<td>Each collection will have an accessions page linked from the root page</td>
</tr>
<tr>
<td>8.</td>
<td>Each collection will contain one or more content pages</td>
</tr>
</tbody>
</table>
| 9.  | Each collection will contain the following policies:  
  - Page look and feel (i.e. CSS, header, footer, left margin, background)  
  - Default access community  
  - Joining policy for the creator community  
  - Leaving policy for the creator community  
  - Joining policy for each access community  
  - Leaving policy for each access community  
  - Formats that will be accessioned |
## 5.2.2 Collection creation

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 10. | The KAS must allow a user to create a new collection. A created collection has  
• a name (which must be unique with respect to the existing collections on the KAS)  
• no creator defined content  
• a root page  
• an accessions page linked from the root page  
• a creator community with the creating user as a member  
• a single access community with no members  
The user must supply a description of the new collection, which forms the root page of the collection. When created, the root page can be edited like any other page.  
Creation of a collection will be logged in the audit trail. |
| 11. | The KAS should allow the capability for the administrator to confirm the creation of a new collection before it is actually created. If provided this function must be capable of being enabled or disabled by the administrator. |

## 5.2.3 Collection maintenance

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.</td>
<td>The KAS must allow the administrator or a member of the creator community to mark/unmark a collection as ‘withdrawn’. When marked as ‘withdrawn’, a collection cannot be accessed by any access community.</td>
</tr>
<tr>
<td>13.</td>
<td>The KAS should allow the administrator or a member of the creator community to mark/unmark a collection as ‘mothballed’. When marked as ‘mothballed’, a collection cannot be modified by the creator community, but may continue to be accessed by the access community (unless the collection has been marked as ‘withdrawn’).</td>
</tr>
</tbody>
</table>

## 5.2.4 Collection deletion

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 14. | The KAS must allow the administrator or a member of the creator community to delete a collection that contains no user defined content. Deleted collections are not longer held on the KAS.  
Deletion of a collection will not be logged in the audit trail. |
| 15. | The KAS must allow the administrator to delete a collection that has been mothballed and the contents archived. Deleted collections are not longer held on the KAS.  
Deletion of a collection will not be logged in the audit trail. |

## 5.2.5 Collection archiving

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 16. | The KAS must allow the administrator to archive a collection. An archived collection is representation of the current state of a collection, including the content, communities, and policies of a collection.  
The archived copy is to be represented as a set of XML documents. All content (e.g. images, sound, video) is to be represented using a suitable long term preservation format. The long term preservation formats will be:  
• Text: XML  
• Images: TIFF, JPEG, or JPEG 2000  
• Sound: MP3  
• Video: MPEG-4  
Archiving of a collection will be logged in the audit trail. |
<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>The KAS must allow the administrator to restore a collection from an archived copy. Restoring of collection will be logged in the audit trail.</td>
</tr>
</tbody>
</table>

5.3 Content

This section details the requirements concerned with creating, deleting, and managing content within the KAS.

Collections contain content. Content is simply information made available by creators to members of the access communities. If a collection is thought of as a web site, content is the pages on that web site.

The KAS, however, does not store web pages (i.e. HTML or XML text). Instead, the KAS stores instructions on how to create web pages, and the web pages are generated automatically when a page is requested by an accessor. In this specification these instructions are known as pages.

A key requirement of KAS is to allow creators to add content even though they may not be skilled in using computers.

Normally, creators create content by entering text and including a simplified form of markup. This markup allows creators to specify headings, lists, emphasised text, captions, and links. Creators can also add objects (e.g. images, video clips, and sound files) to pages. When the page is accessed the content is converted to HTML or XML, combined with the decoration specified as policy for the collection, and served to the accessor.

Alternatively, a creator can create pages by simply uploading content (e.g. an image). The KAS automatically creates a page using the content and information supplied by the creator. The resulting web page is a normal page, and can be subsequently edited by members of the creator community.

Access control model is designed to be flexible and simple. Members of the creator community can always access all pages in a collection. In addition, each page can be associated with one or more access communities. Only members of the associated access communities can access that page.

Every page created is listed on a special accessions page. This page is automatically created when the collection is created and cannot be deleted or edited. The contents of this page is simply a list of every page in the collection, listing date created, creator, last modified, last modified by, and title (which is also a link to the page). The table can be sorted by any of these fields. This allows pages to be created and found (either by browsing the accessions page, or searching) even if the page has not been linked to other pages in the community.

If access to a page is restricted to particular access communities, this page will only appear in the accessions list when viewed by a member of the particular access community. For other accessors the page will be suppressed from the list.

The KAS should provide the facility for creators to have an on-line discussion about a page. It is envisaged that this would take the form of a discussion page associated with each content page and linked either by a hypertext link, or by a ‘tab’ on the content page. The discussion page itself will take the form of a list of discussion topics, each of which will have one or more posts. Creators will have the option of creating a new discussion topic (automatically creating the first post), or responding to an existing post.
### 5.3.1 Root page

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 18 | The KAS must support a root page. This is a special content page. The root page is automatically generated and always contains links to (among other pages):  
  - the accessions register  
  - policy setting pages  
  - simple content search  
  - access community page (only for creators)  
  None of these links can be modified or deleted by creators.  
  The root page also contains normal content. This normal content may be edited just like the content of any other content page. |

### 5.3.2 Accessions register

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 19 | The KAS must support an accessions register. This is a special content page linked from the root page. The accessions register page is automatically generated and contains a list of every page in the collection. For each page the list details the:  
  - date/time the page was created  
  - page title  
  - creator  
  The default sort for the list is by date/time created. A user can request that the page be resorted by page title or creator (in either case the secondary sort is by date/time created).  
  All users can view the contents of the accession page, but the details of pages that do not match the access community of the user are suppressed.  
  The contents of the accessions register page cannot be edited or modified by a creator. |

### 5.3.3 Content creation (basic)

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 20 | The KAS must allow a member of the creator community to create a new page. The creator must enter a page title for the new page. The creator may (optionally) enter a set of access communities to be associated with the page. The KAS will not allow non-existent access communities to be entered. If the creator does not enter an access community, the default access communities are automatically applied.  
  A created page initially has no content, however the creator is immediately put in edit mode to allow them to create the content.  
  The created page is listed on the accessions page.  
  Creation of a page will be logged in the audit trail if pages are not being controlled by a version control system. |
### 5.3.4 Content creation (supported)

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.</td>
<td>The KAS must allow a member of the creator community to upload page content causing the automatic generation of a new page. The page content is a file specified by the creator. The creator must enter a page title for the new page. Some metadata will be captured automatically with the upload, such as the identity of the user, the data/time, and technical information about the object. The user can optionally enter a description of the object (the user will be prompted for the names of participants, location, and date/time). The description forms part of the content of the created page. The creator may (optionally) enter a set of access communities to be associated with the page. The KAS will not allow non-existent access communities to be entered. If the creator does not enter an access community, the default access communities are automatically applied. The created page is listed on the accessions page. Creation of a page will be logged in the audit trail if pages are not being controlled by a version control system.</td>
</tr>
</tbody>
</table>

| 22. | The KAS should support capturing audio and/or video directly from a computer. The requirements will be identical to those given previously, except that the content is captured from the computer instead of being in a file specified by the creator. |

| 23. | The KAS should support capturing content from an email mailed to the KAS. The title of the page will be the subject of the email, and the content will be the content of the email (including attachments). Email will only be accepted sent from accounts associated with members of the creator community. |

| 24. | The KAS may support capturing audio and/or video directly from a phone (particularly a mobile phone). Some metadata will be captured automatically with the upload, such as the identity of the user, the data/time, and technical information about the object. A default title and labels will be applied to the object. |

### 5.3.5 Content alteration

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| 25. | The KAS must allow a member of the creator community to edit the contents of a page. The contents of a page will be a simplified representation of HTML and will consist of text, images, sound, video and markup. The markup will allow:  
- 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> level headings  
- text emphasis (e.g. bold)  
- bulleted lists  
- captions  
- numbered lists  
- hyperlinks  
Images, sound, and video content will be able to be uploaded and added the page.  
Hyperlinks can be expressed either as full URLs, or shortcut references to pages within the collection. The creator will be able to save or cancel the changes. Modification of a page will be logged in the audit trail if pages are not being controlled by a version control system. |

| 26. | The KAS must allow a member of the creator community to edit the metadata associated with a page. The metadata edited is:  
- Title of the page  
- Access communities |

<p>| 27. | The KAS should allow the inclusion of arbitrary HTML in the page |</p>
<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.</td>
<td>The KAS may allow creators to save their changes but not to update the published representation of the page.</td>
</tr>
<tr>
<td>29.</td>
<td>The KAS should record each version of the page as it is saved in a version control system.</td>
</tr>
<tr>
<td>30.</td>
<td>If versions are saved, the KAS should present a list of previous versions of the page on a subpage. The list is to include the date the version was created and the identity of the creator that saved the version. The list is to be sorted from the most recent version to the oldest version.</td>
</tr>
<tr>
<td>31.</td>
<td>If versions are saved, the KAS should allow creators to revert pages to previous version of the page. If a page is reverted, this is considered to be a new version of the page (i.e. the page that has been reverted is the immediately past page in the version history).</td>
</tr>
<tr>
<td>32.</td>
<td>The KAS should allow a creator to lock/unlock a page that they have created (i.e. prevent other creators from editing the page).</td>
</tr>
<tr>
<td>33.</td>
<td>The KAS may support a GUI editing tool to create and modify the content of pages.</td>
</tr>
</tbody>
</table>

5.3.6 Content access

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>34.</td>
<td>The KAS must publish pages on the web using HTML-4.0, XHTML, or XML.</td>
</tr>
</tbody>
</table>
| 35. | The KAS must support the specification of a standard style for the collection (a style policy). This style will contain:  
- A CSS style for the contents of the page  
- A standard background  
- A standard header  
- A standard footer  
- A standard left hand margin  
The elements of the standard style may be uploaded at any time by any creator. Uploaded style elements take effect immediately. |
<p>| 36. | The KAS must publish pages using the style associated with the collection. |
| 37. | The KAS must publish web pages with persistent URLs. Once a page is created its URL will not change (except that the domain name portion may change if the domain is changed). The URL must not contain elements that are derived from the technology used to implement the KAS (so that the URL will remain the same even if the KAS technology changes). |
| 38. | The KAS must allow creators to suppress/unsuppress the publication (access) of pages to accessors. A suppressed page exists and can be viewed/modified by creators, but not by members of access communities. An attempt by an accessor to access a suppressed web page will result in the return of special web page indicating that the requested page has been withdrawn. |
| 39. | The KAS must index all pages. The index is to cover all textual content and the title. When pages are modified the contents are to be re-indexed. |
| 40. | The KAS must provide a simple text search function on the root page. This will simply look for pages that contain any of the words (sequences of characters separated by whitespace). The search results are presented as list on a page. Each entry is to include the page title, which will be a link to the actual page. The search results must be filtered to remove pages that the user cannot access as they are not a member of the appropriate access community. |
| 41. | The KAS may support advanced text search on the root page. This will provide the ability to enter search terms in the following fields: ‘any words’, ‘all words’, ‘exact phrase’, and ‘must not include’. The results will be displayed as for the simple text search. |
| 42. | The KAS must always serve pages to members of the creator community (even if they are also members of access communities that would otherwise not be able to access the pages). |</p>
<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>The KAS must only serve pages to members of access communities that match the labels associated with the page. An attempt to access an unauthorised page will result in a HTTP 404 error.</td>
</tr>
<tr>
<td>44</td>
<td>The KAS may support marking particular content on pages with access labels to allow redaction of content. When such a page is accessed, an accessor will retrieve the content that they are authorised to receive. Content they are not authorised to receive will be suppressed.</td>
</tr>
<tr>
<td>45</td>
<td>The KAS must support the ability for users to identify themselves when attempting to access the collection. This will be by a ‘log in’ page where the user will enter their id and password. Note that users will not select the communities that they wish to log in to; the KAS will automatically log them into all communities (creator and access) that the user belongs to. Members of access communities must not be able to list the access communities associated with a collection.</td>
</tr>
<tr>
<td>46</td>
<td>The KAS must support the ability to log out from the collection.</td>
</tr>
<tr>
<td>47</td>
<td>The KAS must support the ability for the creators to set a robot exclusion policy on the collection or part of a collection.</td>
</tr>
<tr>
<td>48</td>
<td>The KAS should provide a discussion forum for each page. Any user (including members of access communities) able to access the page will be allowed to create comments in the discussion forum. The discussion forum will be linked (via a link or a tab) from the base page. The forum will consist of a list of subject threads (i.e. a list of subjects, with each subject followed by a list of posts on that subject). The title of each thread must be highlighted, and each post must be identified by the date/time posted and the user creating the post. Members of the creator community will be able create new threads (by creating the first post), and respond to existing posts.</td>
</tr>
<tr>
<td>49</td>
<td>The KAS may support needs based logging into the collection. This means that users may access pages that are open without logging in, however a login screen is presented when they attempt to access a page that is restricted or attempt to perform a function reserved for creators.</td>
</tr>
</tbody>
</table>

### 5.3.7 Content deletion

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>The KAS must allow pages with no content to be deleted. An attempt to access a deleted page will result in the return of a special web page indicating that the requested page has been deleted. Deletion of a page will be logged in the audit trail if pages are not being controlled by a version control system.</td>
</tr>
<tr>
<td>51</td>
<td>The KAS must not re-use the URL of a deleted web page.</td>
</tr>
<tr>
<td>52</td>
<td>The KAS may delete any links on other pages to the deleted page. Such modifications are to be recorded in the audit log (or version control system).</td>
</tr>
</tbody>
</table>

### 5.4 Communities

The concept of a community is a key concept in the KAS, and distinguishes it from similar systems (such as Wikipedia). In KAS, a community is simply a group of people that can collectively perform a function. A community could be a formally defined community (e.g. a co-op), a family, a group of like-minded people, or even an individual. Membership of a community is controlled by those responsible for the collection. This is a key difference from some other annotation systems and will be further discussed below.

The two types of community reflect the functions that can be performed. These types are:

- Creator community. This community is responsible for creating content and annotations. It is also responsible for defining and enforcing the collection.
policies. The most important set of policies are those that govern membership of the communities. Each collection has only one creator community.

- **Access community**: Members of an access community may only access content. A collection may have several access communities, each associated with a different set of pages. The access communities are set up by the creator community which gives them meaning. For example, the creator community could set up an access community and only allow initiated men to be members. By allowing the creator community to define the meaning of the access communities we avoid having to pre-define access categories. This approach is quite different to that of Koopman which uses metadata to pre-define a set of access categories.

As already mentioned, the key policies set by the creator community are those controlling community membership. Each community has two policies, one to control joining a community, and the second to control leaving the community.

Supported policies for joining a community will be:

- **Open**. An open joining policy means that anyone may be a member without any formalities. In practice, an open membership would mean that no restrictions are placed on this function (i.e. an open access community means that the content is freely available). The creator community may enforce a policy of recording who becomes a member (requiring the person to supply a name, or recording the host addresses), or it may decide that membership is anonymous.

- **Request**. A request joining policy means that a person must request membership, but that approval is automatic. Essentially this option means that the system records who are members of a community.

- **Approved**. A request joining policy means that a person must request membership and this must be approved. Various approval options will be provided including: approval by any one member of the creator community, approval by a quorum (minimum number), approval by a majority, approval by all, and blackballing (approval withheld if anyone votes against approval).

Any member of a community may leave a community at any time. In addition, the KAS will support policies for when people need to be removed from communities. Supported policies for deciding that a person should be removed from a will be by any one member of the creator community, by a quorum, by a majority, and by all.

The security and access control mechanisms in KAS are not intended to provide the level of protection required to protect secret business. The reference groups felt that the system was unlikely to be sufficiently trusted to hold this type of information.

### 5.4.1 Community management

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>53.</td>
<td>The KAS must provide one creator community for each collection</td>
</tr>
<tr>
<td>54.</td>
<td>The KAS must provide one or more access communities for each collection. All access communities are named.</td>
</tr>
<tr>
<td>55.</td>
<td>The KAS must allow the creator community to specify a default set of access communities (default access policy). These will be applied to every page that does not have a specific set of access communities applied to them. The default set may be modified at any time by any member of the creator community. If the default set is modified, the modified access control will apply immediately to all pages that use the default set. Changes to the default set of access communities are to be recorded in the audit log.</td>
</tr>
</tbody>
</table>

### 5.4.2 Community creation

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
**Id** | **Requirement**
--- | ---
56. | The KAS must allow any member of the creator community to create a new access community. The access community is initially empty. Note that creator communities cannot be created. Creation of an access community is to be recorded in the audit log.

### 5.4.3 Community maintenance

**Id** | **Requirement**
--- | ---
57. | The KAS must allow any member of the creator community to modify the policies associated with any access or creator community. Modification of the policies associated with an access community is to be recorded in the audit log.
58. | The KAS may support various controls for modifying community policies. These are:
- Modify if any creator requests change
- Modify if at least n creators request change
- Modify if at least n% of creators request change
- Modify if all creators request change
This policy should be applied individually to each community policy. Some of these options will require some form of voting. A creator will initiate a change. Other creators will be notified of the pending change and given the opportunity to endorse or reject the change. If sufficient creators agree to the change it is put into effect.

### 5.4.4 Community deletion

**Id** | **Requirement**
--- | ---
59. | The KAS must allow any member of the creator community to delete an access community (note that the creator community cannot be deleted). If the deletion would cause pages to become inaccessible (i.e. because they could only be accessed by members of the deleted community), the user requesting the community to be deleted must select a new community for those pages, or request that those pages be marked as withdrawn. Deletion of an access community is to be recorded in the audit log.

### 5.4.5 Joining communities

**Id** | **Requirement**
--- | ---
60. | The KAS must support a joining policy for each community. This policy can be set by any user of the creator community.
61. | The KAS must support the following joining policies:
- Open (anonymous). Anyone may join the community at any time without supplying any credentials. In practice, this option means that an attempt to access a page controlled by an open access community will be successful irrespective of who the user is (or even if they have identified themselves).
- Request (identified). Anyone may join the community upon request, but they must supply details about themselves. These details include: id and password. Optional details would include a name and contact details.
- Approved. Users may only join the community when approved by a member of the creator community. Potential users must fill in a request form on a special web page which is then approved by a member of the creator community.
<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.</td>
<td>The KAS must support potential users requesting to become members of access or creator communities. This will be via a page that allows the user to enter an id, a password, a name, email address, other contact details, and a short message. After the request has been approved (or denied) the user will receive an email giving information about the decision, and, if approved, about the collection. Requests to join a community are to be recorded in the audit log.</td>
</tr>
<tr>
<td>63.</td>
<td>The KAS must email requests to join communities to all members of the creator community. This email to include the details of the person requesting to join the community and details as to how the creator can approve or decline the request.</td>
</tr>
<tr>
<td>64.</td>
<td>The KAS must support a page that lists all pending requests to join communities. Each entry to include all details supplied by the user in making the request, and the identities of the creators that have approved or rejected the user. The entry to provide the ability for a creator to approve or reject the request and to assign the user to an access community. When the conditions for acceptance or rejection of the request have been fulfilled, the user is to be notified accordingly and, if accepted, the user added to the appropriate access community. Approval, or rejection, to join a community to be logged in the audit log.</td>
</tr>
<tr>
<td>65.</td>
<td>The KAS must allow a description to be associated with the creator community and separate description collectively for the access communities. This description is a text file that is sent to new members of the creator community or any access community to inform them about the collection and its policies.</td>
</tr>
<tr>
<td>66.</td>
<td>The KAS must allow the administrator to add a user to the creator community. (It is expected that this function would only be used where all the existing creators have become inactive). The addition of a user to the creator community to be logged in the audit log.</td>
</tr>
</tbody>
</table>
| 67. | The KAS may support various controls for approving or inviting new community members. These are:  
  - approval by any one member of the creator community  
  - approval by a quorum (minimum number)  
  - approval by a majority  
  - approval by all  
  - blackballing (approval withheld if anyone votes against approval)  
This policy should be applied individually to each community. Some of these options will require some form of voting. The approval process will be initiated by a user requesting to join the community, or by a creator proposing that the user be invited into the community. Other creators will be notified of the proposal and given the opportunity to endorse or reject the candidate. If the acceptance (or rejection) conditions are met, the approval process terminates. |
| 68. | The KAS may support invitations to join communities. In this mode, a creator issues an invitation to join a community and the user accepts or declines the invitation. |

### 5.4.6 Leaving communities

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>69.</td>
<td>The KAS must support a leaving policy for each community. This policy will be set by any user of the creator community.</td>
</tr>
<tr>
<td>Id</td>
<td>Requirement</td>
</tr>
<tr>
<td>----</td>
<td>-------------</td>
</tr>
<tr>
<td>70.</td>
<td>The KAS must allow a user to leave the creator or all access communities at any time. (Note that users are not aware of which access communities they are members of, so a user must leave all access communities at a single time). Leaving a community will be an option on the user’s account maintenance page. The departure of a user from a community will be logged in the audit log.</td>
</tr>
<tr>
<td>71.</td>
<td>The KAS must allow a creator to remove another member of the creator community or a member of an access community. This will be an option on the community list pages. The removal of a user from a community will be logged in the audit log.</td>
</tr>
</tbody>
</table>
| 72. | The KAS may support various controls for removing community members. These are:  
- removal by any one member of the creator community  
- removal by a quorum (minimum number)  
- removal by a majority  
- removal by all  
This policy should be applied individually to each community. Some of these options will require some form of voting. The removal process will be initiated by a creator proposing that the user be removed from the community. Other creators will be notified of the proposal and given the opportunity to endorse or reject the removal. If the removal conditions are met, the user is removed. |

### 5.4.7 Community membership maintenance

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>73.</td>
<td>The KAS must allow users to manage their account. This will be by ‘user management’ page linked from the root page. Clicking on the link (or tab) when logged in will bring up the details of that user. The details will be: id, password, name, email, contact details, and whether the user is a member of the creator community or an access community. All fields (except community membership) will be editable. There will ‘save’, ‘cancel’, and ‘leave’ buttons. Pressing ‘save’ will save any changed details. Pressing ‘cancel’ will discard any changes (as will hitting the back button). Pressing ‘leave’ will cause the user to leave the community. The last is confirmed before being implemented. The modification of a user account will be logged in the audit log.</td>
</tr>
<tr>
<td>74.</td>
<td>The KAS must allow creators to manage access users’ accounts. Creators will be able to select a user and then edit their details just as the user would be able to edit. However, creators will be presented with a list of access communities that user is a member of. The modification of a user account will be logged in the audit log.</td>
</tr>
<tr>
<td>75.</td>
<td>The KAS must allow the administrator to manage access users’ accounts. Creators will be able to select a user and then edit their details just as the user would be able to edit. However, creators will be presented with a list of access communities that user is a member of. The modification of a user account will be logged in the audit log.</td>
</tr>
</tbody>
</table>
5.5 System requirements

5.5.1 Security of contents

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.</td>
<td>The KAS must prevent access to a collection by anyone apart from</td>
</tr>
<tr>
<td></td>
<td>• the administrator</td>
</tr>
<tr>
<td></td>
<td>• creators</td>
</tr>
<tr>
<td></td>
<td>• members of the access communities</td>
</tr>
</tbody>
</table>

5.5.2 Preservation

Over time it is expected that significant effort will be expended in creating annotations and adding other original content. This information will have continuing value and is expected to outlast the KAS itself. Consequently, it is important that the KAS provides the ability to perform long term preservation of the digital material. Functions to be provided include:

• Disaster recovery. To ensure preservation of the material in the face of unexpected catastrophic failures of the KAS, a proper disaster recovery system must be implemented, including off-site back-up on archival quality media. As far as possible this system must be automated to reduce cost and the possibility of administrative error.

• Long term preservation formats. All original content should be normalised upon creation to a small range of formats that can be supported over the long term. Examples of such formats are HTML and JPEG.

• Persistent identifiers. To allow long term persistent access to the information, all pages must be identified using persistent identifiers to ensure that links remain stable.

• Export capability. As already noted, the life of the information stored within KAS will exceed that of the KAS itself. Consequently, it is essential to provide support to easily export the information for transfer to a replacement system. The export function is even more critical in KAS than in a system for larger archive. The operation of the KAS in an organisation over the long term is likely to be subject to the vagaries of funding and interest. It is consequently essential that the information be easily exported and packaged, either for transfer to another organisation, or for ‘mothballing’ until this can be arranged.

Some of these functions have already been covered in other sections.

<table>
<thead>
<tr>
<th>Id</th>
<th>Requirement</th>
</tr>
</thead>
</table>
The KAS must support a full disaster recovery strategy. (Note that implementation of disaster recovery strategy can only be performed by the organisation operating a KAS and so cannot be covered in this specification). The KAS must:

- identify all information that makes up a collection
- make it easy to backup and restore this information

6 Non-functional requirements

These requirements are concerned with how well the KAS performs the functions identified in the previous section.

In the following requirements we distinguish between the KAS software and the complete KAS implementation. The KAS implementation is the particular combination of hardware (e.g. CPU, memory, discs) and software components (e.g. KAS software per se and third party software) supplied to this specification. The KAS software will have higher requirements for its design and implementation than the complete KAS implementation. It is expected that by improving the hardware (and possibly other software components) the performance of the KAS implementation will improve proportionally.

6.1 Response time

The KAS implementation must respond to a request for a web page within 1 second 95% of the time. When the page contains large objects (e.g. image, audio, video), the download speed is to be at least 0.5 Mbit/s.

The KAS implementation must save a modified text only page within 10 seconds 95% of the time. When the page contains large objects (e.g. image, audio, video), the upload speed is to be at least 0.5 Mbit/s.

Response times are to be measured on a computer that is located on the same physical network as the KAS server. The response time for retrieving a large object (e.g. image, audio, or video) is the time taken for the beginning of the object to be received.

6.2 Number of collections

The KAS software must support at least 32,000 collections. The KAS implementation must support at least 10 collections without degradation of performance.

6.3 Number of access communities

The KAS software must support at least 32,000 access communities. The KAS implementation must support at least 20 access communities for each collection without degradation of performance.

6.4 Number of users

The KAS software must not limit the number of unregistered users.

The KAS software must support at least 128,000 user registrations. The KAS implementation must support at least 500 user registrations for each collection without degradation of performance.

The KAS implementation must support at least 100 simultaneous users being logged in without degradation of performance. Performance testing is to assume that each user will request, on average, a page every 10 seconds.
6.5 Collection size

The KAS software must not limit the number of pages in a collection, nor limit the number of objects (images, audio, or video) associated with the pages.

The KAS implementation must support at least 10,000 pages, with at least 2,000 images, 2,000 audio objects, and 500 video objects.

7 Challenges and Development Issues

There are challenges and development issues in regards to the design and possible future creation of KAS, which will need to be addressed. Addressing these challenges and issues are going to require ongoing dialogue between the T&T project and Koorie communities throughout the development process.

As identified by T&T project, trust is an issue that is going to be a major challenge in regards to the type of organisation or agency that will maintain and care of KAS. Data so far is suggesting that a Koorie organisation, such as the Koorie Heritage Trust Inc., would be the preferred type of organisation. This preference is strongly linked to the distrust of how government would treat and access the original records.

Placing the storage and care of KAS within the hands of a Koorie organisation addresses the needs of community control of Koorie oral memory, however it raises issues that may not be faced by government agencies in regards to long-term sustainability of KAS. After any initial funding is exhausted, community organisations usually do not have the ongoing technical expertise, or sufficient funding to maintain such a system. Sustainability issues include:

- Development of the system to add features or fix bugs;
- Funding to operate the system and support the necessary infrastructure;
- Support for communities in using the system – such as training and technical advice;
- Adequate storage facilities for original documentation;
- Technical features that make it difficult to support sustainability – e.g. complex technology, expensive support costs, lack of adequate training within the institution;
- Poor design or implementation that makes it hard for the Koorie communities to collaborate in generating or maintaining the alternative records;
- Quality control of entries.

Community interest and commitment to maintain and develop content – particularly handling the life cycles that on-line communities pass through – is a challenge that will require:

- Achieving buy-in from the Koorie communities;
- Achieving sustained engagement between PROV and the Koorie communities so that links and content are adequately maintained. This will probably require champions within both PROV and the Koorie communities;
- Lack of trust on the part of an archive in providing a forum for critical comment about the records and the story that they tell, where the forum is not under archival control.

Another challenge is access to the necessary infrastructure to use the system by Koorie community members. Access to the Koorie System would require an internet connection, and a certain amount of computer literacy and technical confidence. While both of these requirements are diffusing out into the community, this may be a challenge in less technically supported or rural communities.

References


